BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/911,819	07/24/2001	John T. Micco	04899-046001	6291

7590 08/03/2004

Kevin J. Canning, Esq.
Lahive & Cockfield, LL.P
28 State Street
Boston, MA 02109

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2124	

DATE MAILED: 08/03/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/911,819	MICCO ET AL.	
	Examiner	Art Unit	
	Tuan A Vu	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 July 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-56* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-56* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *24 July 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the application filed July 24, 2001.

Claims 1-56 have been submitted for examination.

Double Patenting

2. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

3. Claims 1, 11, 29, and 39 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claim 16 of copending Application No. 10/190288 (hereinafter '288). Although the conflicting claims are not identical, they are not patentably distinct from each other because of the following conflicts.

As per claims 1 and 29, '288 claim 16 also recites

processing a definition of a function associated with a first language (e.g. *source code for ... more functions in a first programming environment; ...processing the source code to create a component; component comprises... COM object; COM source code files include an Interface Description Language ... class definition and implementation files ...; process IDL files to produce...*- Note: class definition and implementation files are equivalent to definition of a

function being processed) to create description information about the function (Note: IDL reads on description of how the function is specified),

the description being sufficient to enable translation of a call to the function into a call to a corresponding function in a second language (e.g. *component is usable by the application ... second programming environment to access ... functions of the component; converting the source code from a first language to a second programming language; compiling the converted source code files, converted COM source code files and processed IDL files to produce object files*).

But '288 does not recite that the translation using the description is being done 'without requiring processing the one or more functions of the component'. It was known in the art at the time of the application that a definition language file (e.g. IDL) is for setting the specification or language specific declaration for a function, class or method adapted to be used for interfacing in heterogeneous environment in the art of software development. Besides, '288 further recites '*...IDL compile ... component type library file ... wherein processing ... version of the component ... does not include type information*' (see claim 15 or 16). Thus, this lend to the interpretation that by compiling the IDL and associated libraries of the functions, it is not necessary to process the target programming language rules on how the functions are to be defined because the component library file from IDL compilation has provided the sufficient information (IDL , as noted from above); hence, without need to verify of the type information when the functions is implemented in the second language. Thus, '288 has implicitly disclosed translation without requiring processing the one or more functions of the component, i.e. the object files produced from the IDL and related libraries and header files.

As per claims 11 and 39, '288 claim 16 also recites

file of description items (*an Interface Description Language ... class definition and implementation files ...*- Note: class definition and implementation files are equivalent to definition of a function being processed), description associated with a first language and sufficient to enable to translate a call to a corresponding call in a second language; and

using the file of description items to translate a first program call into a second program file (*component is usable by the application ... second programming environment to access ... functions of the component; converting the source code from a first language to a second programming language; compiling the converted source code files, converted COM source code files and processed IDL files to produce object files*); and further includes the implied recitation of 'without requiring processing the one or more functions of the component' as has been analyzed from above.

The claims from claims 1, 11, 29, and 39 are also rejected for being dependent on a rejected base claims.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-20, and 29-48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shannon et al., "Mapping the Interface Description Language Type Model in C", November 1989, IEEE Transactions on Software Engineering, Vol. 15, No. 11, in view of Research Systems, "IDL", copyright 1994 (hereinafter Research_Systems).

As per claim 1, Shannon discloses a method comprising

creating description information or *IDL* about a function in a first language (e.g. *Diana structures ... mapped ...to C, Ada Breadboard Compiler* – Introduction pg. 1333-1334; *class, function* - Fig. 1 pg. 1335 – Note: The conversion based compiler extending Ada structures to C reads on creating a description interface from en existing Ada function or data structures),

the description being sufficient to enable translation of a call to the function into a call to a corresponding function in a second language (*...should be permitted by the C compiler* - Introduction pg. 1333-1334) without requiring processing of the definition of the function (e.g. *... should be natural and not require knowledge of implementation details* – pg. 1333, right column).

But Shannon does not explicitly disclose processing a definition of a function associated with the first language to create the description information. In view of the suggested conversion from one language to another, e.g. from ADA structures to C as noted above, the processing of a source function in order to specify an interface definition language is strongly suggested. Research_Systems, in a method using IDL analogous to Shannon's, discloses mapping to fourth-generation programming language like C, and using such IDL/programming language to implement applications as in Math functions, graph plotting, Image Processing (pg. 1-5 – Note: all of these applications require functionalities that involve and require processing of

input/output data), hence implies processing of each of those application function in order to create an intermediate mapping language, e.g. IDL. In case Shannon does not already teach processing a application function in order to create an description file, it would have been obvious for one of ordinary skill in the art at the time the invention was made to apply the use of the IDL to implement functions as taught by Research_Systems because this interactive language would allow the developer to prototype and assist in developing complete real-world and highly data and processing extensive applications (math, image processing) from one language, e.g. mathematical formulas to another language, e.g. C or Fortran, thus enabling advanced high-level programming language user-friendly constructs and compiling/debugging benefits that come with such language.

As per claims 2 and 4, Shannon discloses a file of description items and derived description information (e.g. sections II, III – pg. 1334-1339).

As per claim 3, Shannon (combined with Research_Systems) discloses using derived information about the function to translate the call to the function into a call to a corresponding function (in a first language) in the second language (e.g. section III pg. 1336; Fig. 3 – Note: C is the second language and Math or graphical functions are in first language). In view of the applications suggested by Research_Systems from above, the examining of the function in those graphical or mathematical functions along with their definition would have been obvious based on the rationale of claim 1 because creating a IDL defining a target language constructs necessary implies the examining of how the source application function (such as mentioned by Research_Systems) is defined, i.e. an inherent step prior to specifying an interface language as intended by Shannon.

As per claim 5, Shannon discloses creation of C language constructs, hence has implicitly disclose the *.lib* files associated with the assembling of object files prior to linking in C. Therefore, Shannon has implicitly disclosed library wherein entries associated with the assembling process in C compiler because at the time the invention was made it was a known concept that C compiler create lib files during a pre-linking compiler process.

As per claims 6 and 7, Shannon does not explicitly disclose processing of the function and deriving a number of declared formal inputs to the functions. But Shannon discloses a declaration with a set of input and output port (ch. B – pg. 1335; Fig. 2, pg. 1336). Official notice is taken that the declaration of formal input and output, and scope of variables declared in a function in 4th generation like C was a known concept at the time the invention was made; hence the IDL input/output provision as suggested by Shannon should imply must-have analysis of the first language input/output formal parameters leading to creation of C formal parameters, i.e. list of input or output/return parameters otherwise the target function declaration would not result in a correct function declaration.

As per claim 8, Namespace and function scope involving local and global parameters are known in 4th generation like C as suggested from the above Official notice. Hence, the analysis of the first language function in order to derive a scope for declaring C function declaration would have obvious based on the implicit teachings as mentioned from the rationale of claims 6 and 7.

As per claims 9 and 10, Shannon (in view of Research_Systems) does not explicitly teach determining of variable arguments in a function and a variable return of results. Official notice is taken the advanced languages like C providing a variable number of arguments, e.g.

input arglist[], or argv[]; or a variable output/return in form of an array, or pointer to a struct or linked list, was a known concept at the time the invention was made. Hence, determining whether a function to have a variable input arguments or return variables would also have been obvious according to the rationale as used in claims 6-8 because complex and highly probabilistic applications as suggested by Research_Systems, math or graphic processing, might entail a non-fixed amount of inputs or outputs, and analyzing such variability of parameters would allow the target code to accommodate for such eventuality, using the known approach provided by C as mentioned above.

As per claim 11, Shannon discloses providing a file of description items, information about a function in a first language (e.g. *Diana structures ... mapped ...to C, Ada Breadboard Compiler* – Introduction pg. 1333-1334), the description being sufficient to enable translation of a call to the function into a call to a corresponding function in a second language (*...should be permitted by the C compiler* - Introduction pg. 1333-1334) without requiring processing of the definition of the function (e.g. ... *should be natural and not require knowledge of implementation details* – pg. 1333, right column); and using the file description items to translate a function into a second program file (see Shannon: Introduction)

But Shannon fails to explicitly disclose using information about a function associated with the first language to translate a first program file into a second program file. But in view of the combined teachings of Shannon/Research_Systems, a function in a first language, like Matlab or ADA functions file, the limitation of converting a first language file into a target file would have been obvious following the rationale as set forth in claim 1.

As per claims 12-13, referring to rationale of claims 6-7, the providing of descriptor for a declared a number of formal inputs or outputs to the functions would also have been obvious because analyzing the first language function necessarily requires implementing an interface language with identification so to enable mapping such function data flow requirements, e.g. input or output number of parameters.

As per claims 14-16, refer to the rationale of claims 8-10.

As per claim 17, Shannon does not explicitly disclose for each call in the 1st program file, retrieving an item from the file of description items, and using information description in the item to translate the first language function into a call corresponding to the 2nd language; and storing the translated function in the second program file. But in view of teachings by the combination using Shannon IDL and function structures and type specifications in light of the application using IDL by Research_Systems in claim 1, the above limitation is implicitly disclosed by Shannon, and would have been obvious according to the rationale of claim 1 and 11.

As per claims 18-19, refer to rationale as set forth in claims 15-16, respectively.

As per claim 20, refer to rationale as set forth in claims 12-13.

As per claim 29, this is the computer-medium product version of claim 1, hence is rejected with the corresponding rejection as set forth therein.

As per claims 30-48, these are the computer product claims corresponding to claims 2-20; hence are rejected with the corresponding rejections as set forth therein respectively.

6. Claims 21-28, and 49-56 are rejected under 35 U.S.C. 103(a) as being unpatentable over Elmroth et al., "A Web Computing Environment for the SLICOT Library", December 2000,

Brite-Euram III, Networks Programme NICONET, in view of Research Systems, "IDL", copyright 1994, further in view of Shannon et al., "Mapping the Interface Description Language Type Model in C", November 1989, IEEE Transactions on Software Engineering.

As per claim 21, Elmroth discloses a method comprising:

providing a library file including functions defined by a first language (e.g. *SLICOT Library, BLAS, LAPACK* – pg. 1, Introduction; *Riccati equations* - Fig. 2-3; ...*uploaded* ... *Matlab files, data files, Latex, Scilab* – pg. 2, pg. 6, Fig. 1, 4 – Note: uploaded matrices in Matlab or Latex format , or Riccati equations read on library files - i.e. files served as input to a library generating tool - defined in one language, all such file integrated into the SLICOT library file framework);

processing the library file to create a function library (e.g. *SLICOT Routines* – Fig. 6) and a description file (e.g. PHP scripts – pg. 6-7), the function library including one or more functions defined by a second language, each function in the function library being translated version of a function in the library file (e.g. *SLICOT routines* – pg. 6; ...*written in C or Fortran* – pg.8: Conclusions and Future Work), and

using the description file to translate a program file from the first language into the second language (e.g. *SLICOT routines, Matlab binaries, PHP scripts* – pg. 7-8 – Note: library files xxxxMD or xxxOD files in conjunction with PHP scripts and converting math functions into m-files read on one or more functions translated from the library file in a second language, i.e. Matlab binaries function files)

But Elmroth does not disclose the description file including description information being sufficient to enable translation of a call to the function into a call to a corresponding function in a

second language without requiring processing of the definition of the function; nor does Elmroth disclose that each call in the program file to a function in the library file is translated into a call to a corresponding function in the second language. Converting math functions into another program like high-level programming language like Fortran or C (analogous to suggestion by Elmroth – pg. 8: Conclusions and Future Work) was a known concept at the time the invention was made. Research_Systems, as set forth in claim 1, discloses definition language (IDL) and mapping various application functions to a fourth-generation programming language like C, and using such IDL file, i.e. description file similar to PHP scripts by Elmroth, to implement applications similar to the Riccati Equations or Matlab files by Elmroth, e.g. Math functions, graph plotting, Image Processing (see Research_Systems, pg. 1-5). The high-level definition description file by Research_Systems was an interactive definition language (IDL) well known at the time the invention was made for converting functions in one language into a corresponding functions in another language, like from Math functions to C program as taught by Research_Systems. As set forth in claim 1, Shannon in a similar approach as Research_Systems, also discloses mapping functions from one language to syntax and constructs implemented in C functions (re claim 1). Hence, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide an IDL as taught by Reseach_Systems and Shannon to enhance the scripts by Elmroth so that the IDL description information is used instead of the PHP scripts, the IDL providing sufficient description information as to enable translation of a call to the function into a call to a corresponding function in a second language without requiring processing of the definition of the function; such that each call in the first language program file to a function in the library file is translated into a call to a corresponding function in the second

language (re claim 1: Shannon). The benefits of using IDL would have been obvious because of the same reasons listed by Shannon: the IDL provides type safe implementation into a high-level programming like C, and further does not require processing of the definition of the function (see Shannon: Introduction, pg. 1333-1334).

As per claim 22, this claim includes the limitation as to translate a call to the function into a call to a corresponding function in a second language as in claim 21; hence is rejected as set forth therein. Further, Elmroth discloses a translated version of each function in the library file (*SLICOT Library, BLAS, LAPACK* – pg. 1, Introduction; *Riccati equations* - Fig. 2-3; *...uploaded ... Matlab files, data files, Latex, Scilab* – pg. 2, pg. 6, Fig. 1, 4- Note: uploaded matrices in Matlab or Latex format , or Riccati equations read on functions in the library file - i.e. files served as input to a library generating tool - defined in one language, all such file integrated into the SLICOT library file framework).

As per claim 23, this claim limitations correspond to those of claim 3; hence are rejected as have been addressed in claim 3, using most of Shannon teachings, mapping of function using a description file derived from examining a function in the first language library file, in light of Research_Systems.

As per claims 24 and 25, these claims correspond to limitations of claim 3, 4 and 11; hence are rejected using the corresponding Shannon's teachings in view of the rationale to combine Elmroth, Research_Systems and Shannon as set forth above.

As per claim 26-28, Elmroth discloses a Web call mapping and converting interface (Fig. 6), while Shannon discloses an interface to check call for error and type violation (the User Interface – pg. 1344). In light of the rationale as to combine the IDL teachings by Shannon with

Elmroth's web interface and PHP scripts, the motivation to provide Elmroth' s Web interface a function evaluation interface as suggested by Shannon would have been for the same reasons as combining Elmroth with Shannon as in claim 21. Further, Elmroth does not specify variable input descriptor, variable output descriptor, descriptor for a known number of input or output arguments as recited in claims 18-20. In view of the rationale to combine Elmroth with the IDL by Research_Systems and Shannon, these claims will be rejected as in claims 18-20 respectively.

As per claim 49, this is the computer-medium product version of claim 21, hence is rejected with the corresponding rejection as set forth therein.

As per claims 50-56, these are the computer product claims corresponding to claims 22-28; hence are rejected with the corresponding rejections as set forth therein respectively.

Conclusion

7. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Pat No. 6,243,856 to Meyer et al., disclosing IDL and conversion of animation functions into Java classes.

U.S. Pat No. 6,446, 137 to Vasudevan et al., disclosing IDL and VRPC library to create stub functions.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

(703) 872-9306 (for formal communications intended for entry)

or: (703) 746-8734 (for informal or draft communications, please label "PROPOSED" or "DRAFT" – please consult Examiner before use)

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA. , 22202. 4th Floor(Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

VAT
July 25, 2004

ANIL KHATRI
PRIMARY EXAMINER